Computer Engineering and Applications Vol. 3, No. 2, June 2014

Words Stemming Based on Structural and Semantic Similarity

Mohammad Hassan Diyanati*, Mohammad Hadi Sadreddini, Amir Hossein Rasekh, Seyed Mostafa Fakhrahmad, Hossein Taghi-Zadeh,

Computer Science and Engineering Department, Shiraz University, Shiraz, Iran
**dianati@cse.shirazu.ac.ir, {sadredin, ahrasekh, fakhrahmad}@shirazu.ac.ir, h-taghizadeh@cse.shirazu.ac.ir*

ABSTRACT

Words stemming is one of the important issues in the field of natural language processing and information retrieval. There are different methods for stemming which are mostly language-dependent. Therefore, these stemmers are only applicable to particular languages. Because of the importance of this issue, in this paper, the proposed method for stemming is aimed to be language-independent. In the proposed stemmer, a bilingual dictionary is used and all of the words in the dictionary are firstly clustered. The words' clustering is based on their structural and semantic similarity. Finally, finding the stem of new coming words is performed by making use of the previously formatted clusters. To evaluate the proposed scheme, words stemming is done on both Persian and English languages. The encouraging results indicate the good performance of the proposed method compared with its counterparts.

Keywords: Natural Language Processing, Stemming, Word Similarity, Clustering.

1. INTRODUCTION

In linguistics, stem is the integrated form of words achieved from similar morphology [1]. Therefore, stemming is reducing various forms of words to achieve a common morphological that is called the stem [2]. For example, in the Persian language, the stem of both words "درخت" (tree) and "درختها" (trees) is "درخت" (tree) and in English, two words "small" and "smaller" are stemmed to a common word "small". Of course, it should be noted that stemming is used to categorize the words in groups of similar structures. Therefore, in stemming, words that have the same meaning but different structures are not in the same category. For example, in Persian, the pair of words "مکانها" (locations) and "محلها" (locations) have the same meaning, while stemming algorithms will return two different words "مکان" (location) and "محل" (location) as the stems of these words. Similarly, in English the stems of two words "locations" and "places" are different but they have the same meanings.

Today, with advances in computer-aided language processing stemming has got a wide range of applications in various fields of natural language processing. Due to

the importance of this topic, several algorithms have already been developed to achieve the stem of words.

The main approaches to stemming include structural methods (removed affix), statistical methods and lookup table methods [3].

The structural methods are dependent on the structure of the language morphology. In these methods, to get the stem of a given word, the prefix and suffix of the word are removed based on a set of specific rules. An example for these algorithms is the Porter algorithm [4]. This algorithm has 5 stages. In order to achieve stem of the word at each step, suffixes of the word are removed according to a number of predetermined rules.

In the lookup table methods, each word as well as its stem is stored in a structure, and subsequently these structures are used to find the stem of words. Generally, these methods have a high accuracy for stemming. However, it should be noted that these methods need a lot of space to store the words. Moreover, the lookup table must be updated for each new word.

The statistical methods use a corpus for obtaining construction rules of words. In these methods, the rules will be extracted from the corpus by considering the changes of the words that have the same stem. Some of the existing statistical methods are: Frequency Count, N-gram [5], Link Analysis [2], and Hidden Markov Models [6].

For each of the three main approaches to stemming, different stemming algorithms have been proposed for different languages. Unfortunately, most of these algorithms are language-dependent and based on the structure of a particular language and thus cannot be applied to other languages.

This paper presents a new method for obtaining the stem of words which can be used for different languages. In this method, the stemming task is performed using a bilingual dictionary. As the first step, the words are clustered based on structural similarity and then another clustering is performed based on the semantic similarity. Finally, words stemming is accomplished by making use of the resulting clusters.

2. RELATED WORKS

The Stemming of words is used in natural language processing as well as in some other fields, such as information retrieval. Due to the importance of this issue, much research has already been performed in various languages.

Porter stemmer was presented in 1980 [4]. This stemmer is a reducer stemmer for English language. This algorithm is able to identify the suffixes of words and doesn't pay attention to prefixes. Porter consists of five steps and in each step there are some special rules for removal of the word prefixes.

In [7], Krovetz has offered a stemming method which uses a set of morphological rules and a dictionary to find the stem of words. The Krovetz algorithm is useful for languages in which the construction of words is regular. Unlike Porter, Krovetz is able to identify the prefixes of words in addition to the suffixes.

An unsupervised learning approach to building a non-English (Arabic) stemmer is presented in [8]. This stemming model is based on statistical machine translation and uses an English stemmer and a small (10K sentences) parallel corpus as its training resources. This stemmer is applicable to any language that needs affix removal.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Kazem Taghva introduced a method for stemming which is very similar to Porter. The method can be used just for the Persian language. In this method, a series of morphological rules are used to find the stem of words. Taghva's stemmer is able to remove the prefixes of words in addition to removing suffixes [9].

Sharifloo presented a bottom up approach for finding the stems of Persian words [10]. This method is based on morphological rules and capable to reorganize without changing the implementation. The experiments show that this algorithm has encouraging results in stemming.

In [11], a stemming algorithm based on co-occurrence of words in a corpus has been proposed. The algorithm has been proposed for text information retrieval. The algorithm uses the statistics collected on the basis of certain corpus analysis based on the co-occurrence between two word variants. This stemmer uses a very simple co-occurrence measure that reflects how often a pair of word variants occurs in a document as well as in the whole corpus. The results show that this stemmer can be used as a better alternative to the rule based stemmers.

The stemmer presented in [12] uses a structural approach for stemming of Persian words. For this purpose, it uses the structure of words and morphological rules of the language to recognize the stem of each word. The rules are written based on the morphology of Persian language and its word derivation structure. This stemmer focuses on stemming of nouns, adjectives and adverbs but doesn't pay attention to verbs, because its goal is to improve the performance of information retrieval systems, with respect to this fact that most queries do not contain verbs.

In [13], five different algorithms have been proposed to improve Arabic stemmers. The proposed algorithms were assessed by using more than 1450 Arabic words including different set of affixation, patterns, as well as hollow verbs and various types of strong verbs.

In [14] an unsupervised method of stemming has been proposed which is hybridized with partial lemmatization for Hindi. The stemmer is unique in that it exploits a novel grouping criteria and aims to improve the unsupervised stemming approach. This concept to unsupervised stemming yielded significant improvements in the desired results, when compared to other prevailing approaches of its genre.

3. THE PROPOSED METHODS

In this paper, we propose stemmer that performs word clustering based on structural and meaning similarity of words. As the first step, a bilingual dictionary is used and then clustering is done on this dictionary and consequently, the stem of each cluster is obtained. Finally, stem of a new coming word can be achieved by using the pre-constructed clusters.

The steps of the proposed stemmer are shown In Figure 1.

```
1: Cluster(dictionary)
2: {
3: For each word in Dictionary do
4:   W=selected one word from dictionary
5:   For each word in Dictionary do
6:     If(word structure is similar to W)
7:       Add word to similar_structure_list
8:   For each word in similar_structure_list do
9:       If(word meaning is similar to W)
10:         Add word to cluster_list
11:          Delete word from similar_structure_list
12:         Delete word from Dictionary
13:    Find the stem of cluster_list
14:   Add cluster_list and its stem to cluster_table
15: Return cluster_table
16: }
```

a. Offline part

```
1: Stemming(input)
2: {
3: New_Word=input
4: For each stem in cluste_table do
5:   If(stem is similar to New_word)
6:    Add stem to similar_stem
7: If(similar_stem is empty)
8:  Word_stem=New_word
9: Else
10:  Word_stem= longes stem from similarstem
11: Return Word_stem
12: }
```

b. Online part

FIGURE 1. Steps of proposed algorithm

As shown in Figure 1, the proposed method is divided into two general parts.

In the first part, the whole words of the dictionary are clustered; while in the second part, the stem of a new word is discovered.

Since the clustering of dictionary words is a time-consuming task, this part of the stemming is done offline, and just the part of obtaining stem of new words is performed online. In the next sub-sections, different parts of the proposed stemmer will be described.

3.1. SELECTING A WORD FROM THE DICTIONARY

In this step, a word from the dictionary is chosen. It seems better to select the smallest word containing more than two characters from the dictionary. It should be noticed that word stems usually have at least three characters and hence it is better to start clustering with smaller words.

3.2. SELECTION SIMILAR WORDS

In this part, all words which are similar to the selected word are extracted from the dictionary.

The point that should be considered in this section is that, there are different standard measures for computing words' similarity. In the following parts, we will

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

review some of the measures and finally, a method to determine the similarity of two words will be offered.

3.2.1. COSINE METHOD

In this method, in order to find the similarity of two words, the letters that are common in both words are counted and then it is divided into the total number of letters. This approach is done regardless of the order of the letters [15]. For example, in Persian language, for the pair of words "فیل" (elephant) and "لیف" (fiber) the Cosine similarity would be 100% and for two English words "part" and "trap", the Cosine similarity will be 100%. Because the letters of the two words are all the same and just the location of the letters are different. In Figure 2, the steps of measuring Cosine similarity are shown.

```
1: Cosine(A , B)
2: {
3: For each letter k in A or B  do
4:  If(k in both A and B)
5:   Di=1
6: Else
7:  Di = 0
8: Similarity= SUM(D)/(|A| +|B|)*100
9: Return Similarity
10: }
```

FIGURE 2. Cosine algorithm

3.2.2. JARO METHOD

In Jaro method the same letters of two words are counted and then the numbers of displacement between letters are calculated. Finally, according to the algorithm in Figure 3, the similarity of two words will be computed [16]. For example, in Persian language, Jaro similarity for two words "فیل" (elephant) and "لیف" (fiber) is 88% and in English language the words "part" and "trap" are 50% similar.

```
1: Jaro(A , B)
2: {
3: C = number of Common letters
4: T = number of Transpositions
5: Similarity = 1/3 (C/|A| + C/|B| + (C-T)/C)*100
6: Return Similarity
7: }
```

FIGURE 3. Jaro algorithm

3.2.3. LEVENSHTEIN METHOD

In order to compute the similarity of two words, the Levenshtein method calculates the minimal changes to convert a word to another word. These changes may include removal of a letter, insertion of a letter or replacing two letters [17]. As an example, Levenshtein similarity for two Persian words "فیل" (elephant) and "لیف" (fiber) is 33% and the similarity between two English words "part" and "trap" is 0%. In Figure 4, the steps of Levenshtein algorithm are shown.

```
1: Levenshtein (A ,B)
2: {
3: Create matrix Similarity[|A| , |B|]
4: Initialize Similarity = 0
5: For each letter Ai do
6:  For each letter Bj do
7:    If (Ai == Bj)
8:     Di,j= 0
9:    Else
10:     Di,j = 1
11:  Similarity (i , j) = MIN ((Similarity (i – 1 , j) + 1),
(Similarity (i , j - 1) + 1), (Similarity (i – 1 , j - 1) + Di,j))
12: Return Similarity (|A| , |B|)/(|A| +|B|)*100
13: }
```

FIGURE 4. Levenshtein algorithm

3.2.4. HAMMING METHOD

In Hamming method for computing the similarity between two words, the number of letters matching is counted and then it is divided by the length of the larger word [18].

Therefore, the Hamming similarity for two Persian words "فیل" (elephant) and "لیف" (fiber) is 33% and for two English words "part" and "trap", the Hamming similarity will be 0%.

Figure 5, shows the steps of the Hamming algorithm.

```
1: Hamming(A , B)
2: {
3: For each letter Ai and Bi do
4:  If (Ai == Bi)
5:   Di= 1
6:  Else
7:   Di = 0
8: Return SUM(D)/MAX((|A| , |B|)*100
9: }
```

FIGURE 5. Hamming algorithm

3.2.5. PROPOSED METHOD TO MEASURE THE WORDS SIMILARITY

The chosen method to measure the similarity of two words is based on the maximum matching in number and order of the letters. On the other hand, only is considered the length of smaller word, because in the stemming issue words with different length may have similar stem. For example, two Persian words "درخت" (tree) and "درختها" (trees) have the same stem but their length is different.

The proposed algorithm to measure the similarity of two words is given in Figure 6.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

```
1: Words_similarity(A , B)
2: {
3: If(|A| > |B|)
4:  Swap(A,B)
5: For each i< (|A| - |B|)  do
6:  For each leters B do
7:   If(Bj== Ai+j)
8:    Di++
9: Similarity= MAX(D)/(|B|)*100
10: Return Similarity
11: }
```

FIGURE 6. Proposed algorithm

In Table 1, the similarity of different pairs of English and Persian words obtained by Different methods are given.

TABLE 1.
Compare Similarity Measures

Word1	Word2	Cosine	Leven.	Jaro	Hamming	Our method
فیل	لیف	100%	33%	88%	33%	33%
کتاب	کتابها	89%	66%	88%	66%	100%
برابر	نابرابر	86%	71%	80%	0%	100%
part	trap	100%	0%	50%	0%	25%
tree	trees	86%	80%	93%	80%	100%
use	reuse	86%	60%	70%	0%	100%

According to Table 1, Cosine algorithm is not good for stemming, because in this algorithm similarity of two words with different stems is 100%. On the other hand, hamming algorithm is not suitable for stemming, because in this algorithm similarity of two words with the same stems is low.

As can be observed, in the proposed method, similarity of two words with the same stems is high, and similarity of two words with different stems is low. Thus, this method is suitable for stemming of words.

After selecting the appropriate method for determining similarity of two words, the clustering of dictionary words is performed based on structural similarity.

As an example, in Persian language, in this step, for the given word "کوچک" (small), the set of words "گوچ" (migration), "چک" (check), "کوچک" (small), and "کوچکتر" (smaller) are selected from the dictionary and for the English word "teach", the words "tea" , "teach" , "teacher" , and "each" are selected.

3.3. CLUSTRING BASED ON SEMANTIC SIMILARITY

In this step, clustering is performed based on the meaning of the words.

This time, the words selected in the previous step are placed in the same cluster if they have the similar meaning; otherwise they should be located in different clusters.

In Table 2, different words that had been selected in the previous step are shown.

TABLE 2.
Similar words with their means

Persian words	
Word	Meaning
کوچک	Small
کوچکتر	Smaller
کوچ	Migration
چک	Check
English words	
Word	Meaning
Each	هر
Tea	چای
Teach	معلمی و یا تدریس کردن
Teacher	معلم

As observed in Table 2, the Persian words "کوچک" (small) and "کوچکتر" (smaller) which have similar meaning are placed in the same cluster and the words "کوچ" (migration) and "چک" (check) are not included by this cluster. Similarly, the English words "teacher" and "teach" are in the same cluster while the other two words are not in this cluster because their meaning is different.

3.4. SET THE STEM FOR EACH CLUSTER

After the process of word clustering, a stem for each cluster is determined. The stem of each cluster is the largest substring that is common between words located in the cluster for example, for the cluster containing the Persian words "کوچک" (small) and "کوچکتر" (smaller), the word "کوچک" (small) is selected as the stem of the cluster and for the cluster including English words "teacher" and "teach", the word "teach" is the cluster stem.

3.5. NEW WORD STEMMING

Now, a set of different clusters of words have been created based on structures and meanings. These clusters can then be used to determine the cluster of a new coming word. For this purpose, the stem of clusters that are similar to the new word is selected and thus the stem of word is determined. Two major problems may arise in this stage.

Firstly, it may be a new word not similar to any stem of clusters. In this case, the new word itself will be selected as the stem.

Secondly, the new word may have two or more similar stems. In this case, the longest stem will be chosen as the stem of the new word. For example, in Persian language, if we want to find the stem of "کوچکترین" (smallest), we see that the two words "کوچک" (small) and "کوچ" are selected as the stem but the word "کوچک" (small) is longer and so it is chosen as the stem of "کوچکترین" (smallest).

4. EXPERIMENTAL RESULTS

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

In order to evaluate the presented stemmer in this paper, stemming for both Persian and English languages was performed.

At the beginning a Persian to English dictionary was used. After that, the clustering of dictionary words was performed based on the structural and semantic similarities. The evaluation of the proposed method was conducted. Since for evaluation of the proposed method, a set of words with the stem is required, totally 1,250 words with their stems were extracted from the text corpus of PerTreeBank [19].

After selecting the set of words, the process of stemming was performed using the proposed approach as well as the Taghva stemmer (one of the most famous stemmers for Persian language). Eventually, the accuracy of these methods was investigated.

The results obtained from these two methods are shown in Table 3.

TABLE 3.
Result of the Persian stemming

	Taghva stemmer	Proposed method
No. of words	1250	1250
Correct stem	767	879
Accuracy	61.36%	70.32

As shown in Table 3, the accuracy of the proposed method for stemming of Persian words is better than the Taghva method. It is mainly related to the unnecessary affix eliminations done by the Taghva algorithm. On the other hand, in Taghva algorithm due to lack of use of the dictionary, the correctness of the stems cannot be checked. For example, by Taghva algorithm the stem obtained for the word "فراوان" (abundant) was " فراو", while the correct stem is " فراوان" (abundant). In this example, The Taghva stemmer assumes "ان" as a suffix of "فراوان" (abundant) while "ان" is a part of the word "فراوان" (abundant).

In order to evaluate the proposed method on English language, we used an English-Persian dictionary. Then we selected a set of 1,250 words from data set sortedtest.txt1. Finally, the word stemming was performed using our approach as well as the porter stemmer (one of the most famous stemmers for English language). The results can be seen in Table 4.

TABLE 4.
Result of the English stemming

	Porter stemmer	Proposed method
No. of words	1250	1250
Correct stem	824	869
Accuracy	65.92%	69.52%

From the results given in Table 4, it can be observed that the proposed stemmer is more accurate than Porter. It is mostly due to unnecessary suffix eliminations done

[1] http://www.comp.lancs.ac.uk/computing/research/stemming/Links/resources.htm

by porter. For example, porter eliminates "s" of the word "yes" but it is wrong. The second reason is that the Porter stemmer doesn't remove prefixes of words. For example Porter doesn't remove the prefix "ir" from the word "irregular".

5. CONCLUSION

In this paper, we have presented a method for stemming of words that can be used in different languages. This stemmer uses a dictionary to find the stem of words.

In the first step, the clustering of dictionary words is done based on both structural and semantic similarities. Then the stem of each cluster is selected as a representative of the cluster. Finally, these clusters and their stems are used to identify the stems of new coming words. In the proposed method there is no need to structural knowledge of the language in order to identify word stems. Making use of a dictionary, we can find the stem of new coming words.

Indeed, the proposed method is language independent and can be used for different languages. On the other hand, the clustering of words can be done offline. Therefore the response time of finding the stem of a new word is highly reduced.

REFERENCES

[1] W. B. Frakes and R. Baeza-yates, "Information Retrieval: Data Structures & Algorithms," Information Retrieval, vol. 152, no. 3, p. viii, 504 p., 1992.
[2] M. Bacchin, N. Ferro, and M. Melucci, "Experiments to evaluate a statistical stemming algorithm," Working Notes for CLEF, pages 161-168, 2002.
[3] C. Bento, A. Cardoso and G. Dias, Eds., "Progress in Artificial Intellegence," 12th Portuguese Conference on Artificial Intelligence, pp. 693 –701, 2005.
[4] M. E. Porter and J. Leo, "Competitive strategy: Techniques for analysing industries and competitors Porter, Michael E. Free Press (Macmillan), New York, 396 pages, 17.95," Industrial Marketing Management, vol. 11, no. 4, pp. 318–319, 1982.
[5] J. Mayfield and P. McNamee., "Single N-gram Stemming," In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 415-416, 2003.
[6] M. Melucci and N. Orio, "A novel method for stemmer generation based on hidden markov models," in Proceedings of the twelfth international conference on Information and knowledge management, 2003, pp. 131–138.
[7] R. Krovetz, "Viewing morphology as an inference process," in Artificial Intelligence, vol. 118, no. 1–2, R. Korfhage, E. M. Rasmussen, and P. Willett, Eds. ACM, 1993, pp. 191–202.
[8] M. Rogati, S. McCarley, Y. Yang, "Unsupervised Learning of Arabic Stemming using a Parallel Corpus", ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pp. 391-398 Vol. 1, 2003.
[9] K. Taghva, R. Beckley, and M. Sadeh, "A stemming algorithm for the Farsi language," International Conference on Information Technology Coding and Computing ITCC05 Volume II, vol. 1. IEEE, pp. 158–162 Vol. 1, 2005.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

[10] A. A. Sharifloo and M. Shamsfard, "A bottom up approach to Persian stemming," in Proceedings of the Third International Joint Conference on Natural Language Processing, 2008.
[11] J. H. Paik, D. Pal, and S. K. Parui, "A novel corpus-based stemming algorithm using co-occurrence statistics," in Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, 2011, pp. 863-872. A Structural Rule-based Stemmer for Persian
[12] E. Rahimtoroghi, H. Faili, and A. Shakery, "A structural rule-based stemmer for Persian," *2010 5th International Symposium on Telecommunications*. IEEE, pp. 574-578, 2010.
[13] S. R. El-Beltagy and A. Rafea, "An accuracy-enhanced light stemmer for arabic text," ACM Trans. Speech Lang. Process., vol. 7, no. 2, p. 2:1--2:22, Feb. 2010.
[14] D. Gupta, R. K. Yadav, and N. Sajan, "Article: Improving Unsupervised Stemming by Fusing Partial Lemmatization Coupled with," International Journal of Computer Applications, vol. 38, no. 8, pp. 1-8, Jan. 2012. C. P. Spaulding, "Sine-Cosine Angular Position Encoders," IRE Transactions on Instrumentation, vol. PGI-5. 1956.
[15] C. P. Spaulding, "Sine-Cosine Angular Position Encoders," IRE Transactions on Instrumentation, vol. PGI-5. 1956.
[16] M. A. Jaro, "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida," Journal of the American Statistical Association, vol. 84, no. 406, pp. 414-420, 1989.
[17] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics Doklady, vol. 10, no. 8, pp. 707–710, 1966.
[18] R. W. Hamming, "Error detecting and error correcting codes," Bell System Technical Journal, vol. 29, no. 2, pp. 147–160, 1950.
[19] M. Ghayoomi, "Bootstrapping the Development of an HPSG-based Treebank for Persian," Linguistic Issues in Language Technology, vol. 7, no. 1, 2012.